An Improved RSA Variant

Seema Verma, Deepak Garg
Department of Computer Science, Thapar University, Patiala, India

Corresponding Author Email: seemaknl@gmail.com

Abstract

As RSA is the most popular and widely used in e-commerce, there is the need to make it more and more efficient. One of the RSA variant, i.e., Dual RSA, is designed to reduce the memory consumption for the two RSA instances. Based on Dual RSA Small d, a new scheme is designed such that the online encryption time becomes almost negligible having the same decryption performance as in Dual RSA small d. The scheme is implemented to reflect the theoretical results. The resulting scheme is efficient in encryption, decryption performance and memory consumption. Hence the scheme is suitable to be used in resource constrained environment. Scheme also exhibits the property of semantic security.

Keywords: *Cryptography, encryption, public key*

1. Introduction

RSA cryptosystem [1] was designed in 1977 by Ron Rivest, Adi Shamir and Leonard Adlemen. It is very popular public key algorithm because of its simplicity. The security lies on computing the factors of a large composite integer. Currently factoring 1024 bits integer is assumed to be as complex as workload of 2^{80} which is the current benchmark used in cryptography. The high computational cost and memory usage bring the algorithm in research area. Work is done on various parameters to improve the efficiency of the algorithm. Work on RSA is categorized according to various factors. The improvement over decryption speed is done in variants [2, 3, 4, 5, 6]. Some areas have the requirement of less computational effort on decryption, while the others have the requirement of less computational effort on encryption side. Both the sides of encryption and decryption need to be balanced in many applications, the variants to balance both the sides can be found in [7, 8]. Other factor in RSA cryptosystem is the memory consumption. For the security purpose, the key size needs to be

large, the variants related to reduction in memory consumption is described in [9, 10]. In this paper the information about the work related to computational and memory consumption overhead is given. The paper is structured as follows: second section describes the related study. The proposed work is given in section three. Security analysis and implementation details are given in section four and five respectively.

2. Basic RSA

In RSA cryptosystem [1], the key generation system generates public key (e, N) for encryption and (d, N) for decryption operation. The keys are calculated by taking two random prime numbers p and q of approximately equal sizes to get N=p*q. Then Euler's totient function Φ(N) is computed by calculating (p-1)*(q-1). Public key factor e is chosen as a random odd integer less than Φ(N) such that gcd(e, Φ(N))=1. Then secret component d can be computed using the congruence equation, i.e. RSA key equation; $ed \equiv 1 \bmod \Phi(N)$. If X wants to send a message M to Y, he/she first converts the message to a number less than N. Then X looks up the public key of Y and sends the message to Y by computing $C= M^e \bmod N$. Receiver Y decrypts the received data by computing $M= C^d \bmod N$. The RSA algorithm will work correctly if the plaintext M is relatively prime to the modulus,otherwise factorization of the modulus is revealed. Also gcd(C, N) gives the information of the multiple of the prime factor.

2.1 RSA CRT

Quisquater and Couvreur [2] introduced this variant in 1982. RSA CRT was designed to improve the speed of decryption method of RSA cryptosystem. With this method the decryption is done with two smaller secret keys (d_p,d_q) instead of one large d. The two secret components can be calculated by calculating (d mod p-1) and (d mod q-1) for d_p and d_q respectively. The encryption method is same as that for basic RSA. Decryption method includes the computation of two factors; $M_p=C^{dp} \bmod p$ and $M_q=C^{dq} \bmod q$. The plaintext can then be retrieved by combining the two factors M_p and M_q using Chinese remainder theorem (CRT).

2.2 Dual RSA

In 2007, Sun et al. proposed a variant, Dual RSA [10], to reduce the memory requirement. Dual RSA generates the same public and private exponent for two distinct RSA instances. The variant is useful mainly for blind signatures and authentication/secrecy.

Dual RSA has three schemes in [10], namely Small-e, Small-d, and Generalized Dual RSA. The encryption and decryption methods are the same for all the schemes. The encryption algorithm is same as standard RSA and decryption is done using the CRT method.
Here we are concerned about only one scheme, i.e,, Dual RSA Small d.

Key Generation Algorithm

Here $n_d < n/2$,

1. Select random number x_1 and x_2 with bit size n_d-bit and $(n/2-n_d)$ respectively, if $p_1=x_1x_2+1$ is not prime repeat for other primes.
2. Select random number y_2 and y_1 of bit size $(n/2-n_d)$- bit and n_d bit respectively. if $p_2=x_1y_2+1$ and $q_1=y_1y_2+1$ is not prime chose another prime.
3. The private exponent d is selected randomly such that $gcd(x_1x_2y_1y_2,d)=1$. Calculate e and k_1 with RSA equation $ed=1+k_1(p_1-1)(q_1-1)$
4. Calculate $q_2=k_1x_2+1$, if q_2 is not prime then go to the previous step.
   $N_1=p_1q_1$, $N_2=p_2q_2$, $k_2=y_1$

Encryption is done in the same way as in the standard RSA and decryption is done in the same way as in RSA CRT.

2.3 DRSA

DRSA by Pointcheval [14] is an RSA variant which is based on "dependent RSA problem". Some improvement is given in [15] by Padhey. Three variants are given which are proved to be semantically secure. DRSA is based upon the problem of Computational Dependent RSA, Decisional Dependent RSA and Extraction Dependent RSA. In Computational Dependent-RSA (C-DRSA), the problem is to determine the value of $(K+1)^e \bmod N$ for the given value of $K^e \bmod N$ where K is randomly chosen element in Z_n^*. This problem is as hard as the RSA problem. The DRSA encryption scheme is proved to be semantically secure against chosen-plaintext attacks. In DRSA, the key generation is same as in standard RSA.For encryption method , a random number K is selected that is coprime to the modulus N, i.e., $K \in Z_N^*$. Calculate $A = K^e \bmod N$ and $B = M \times (K+1)^e \bmod N$. The pair (A,B) is sent as the cipher text.

For decryption algorithm, first calculate the value of K as: $K = A^d \bmod N$ and then the message is decrypted by calculating $M = B/(K+1)^e \bmod N$

4. Proposed Work

In this section a scheme is proposed which results in efficient encryption, decryption and memory usage. For this purpose Dual RSA small-d [10] is combined with DRSA [14]. The key aspects of both the variants are combined to form a new scheme.

4.1 Proposed Method

Following is the detail of the proposed scheme. In the proposed scheme the key generation is done as in Dual RSA small-d and idea of encryption/decryption operations is from DRSA.

Key Generation:

- Select the private exponent d with n_d bit (here $n_d > n/2$)
- Calculate e as inverse of the private exponent that comes out to be very large of the order of the modulus.
- The prime numbers p_1, q_1, p_2, q_2 are chosen such that the private and public exponents (d and e) are same for the two moduli, $N_1=p_1*q_1$ and $N_2=p_2*q_2$.

Thus the parameters generated are e, d, N_1 and N_2. For two instances the public and private exponents are calculated as common for both, resulting in less memory consumption.

Encryption Method

The encryption is done in two parts; part 1 is executed any time when server is offloaded and part 2 is executed when the message is received for encryption. Here the modulus N_1 is shown in the computations.

Part 1

The following statements can be calculated offline (before encrypting the message).

Select R as any random number $R \epsilon Z_n^*$.

Calculate $C'=(R-1)^e \bmod N_1$ and

Calculate $R^{-1} \bmod N_1$

Part 2

To encrypt any plaintext M ,

Calculate $C=M*R^{-1} \bmod N_1$

(C', C) is the required cipher text.

Decryption Method

To decrypt the cipher text (C', C),

Calculate $R=C'^d \bmod N_1+1$

Calculate the message $M=C*R \bmod N_1$

4.2 Efficiency

In encryption operation, R^{-1} and C' both can be computed well in advance. For the calculation of C_2, just modular multiplication is required instead of exponent calculation, which is very less time consuming as compared to exponentiation.

The decryption cost is almost same as standard RSA but with small decryption exponent. Here the only extra cost is modular multiplication that is incurring not much cost. In Dual RSA Small-d [10], the encryption exponent e is taken very large; hence the efficiency of encryption process becomes very slow. In the proposed method, since the complex computation can be done in advance (offline calculation), therefore the online encryption requires only one multiplication modulo N_1. This shifting of calculation from online to offline results in very fast online encryption. Also the decryption side of the proposed scheme is computationally as expensive as the Dual RSA Small-d due to only one extra multiplication modulo N_1.

Due to the use Dual RSA small-d key generation memory consumption is low. For two instances of RSA only one pair of public and private exponents needs to be stored.

4.2 Security Analysis

According to the security constraints defined in dual RSA: $n_d > n/3$, $n_d > 341$ (for moduli $= 1024$ bits) and $n_d > 682$ (for moduli $= 2048$ bits)

The study regarding Dual RSA in Hinek [17] and Sarkar and Maitra [18, 19] proved that any other attack doesn't apply to Dual RSA Small-d. Hence the scheme is secure with the present scenario.

The proposed scheme is semantically secure against chosen plaintext attack. Semantic security is equivalent to indistinguishable chosen plaintext attack (IND-CPA). For a cryptosystem to be semantically secure probabilistic encryption algorithm and deterministic decryption algorithm are required. For probabilistic encryption algorithm, randomization must be used. With the use of randomization, by knowing any (plain text, cipher text) pair, an adversary couldn't predict any function that can result in knowing the value of cipher text for any other unrelated plain text. No deterministic public key encryption gives message indistinguishability.

RSA is semantically secure with OAEP [12]. The proposed scheme is the combination of Dual RSA Small-d and DRSA. DRSA as described in the previous section (2), was introduced by Pointcheval [14] in 1999. He proved DRSA and its variants to be

semantically secure against indistinguishability of encryption against chosen plaintext attack and adaptive chosen-ciphertext attacks. Basic DRSA is semantically secure against chosen plaintext attack. In the proposed scheme basic DRSA is used. Pointcheval proved that the DRSA encryption scheme is semantically secure and the DRSA problem is intractable with e greater than 2^{60}. In the proposed scheme, e is much larger than this factor; hence this scheme follows the constraints of the standard model proposed by Pointcheval. For the proof of the semantic security, reader may refer to [14].

In the proposed scheme, to determine any information about the plaintext from the cipher text (C', C) attacker need to have some information about R^{-1}mod N, where R is randomly chosen element. The only way to ascertain any information about the value of R^{-1}mod N_1 is to first compute R. It is not possible without knowing the secret key d. The concept of complete randomization is practical according to [16]. Hence the proposed scheme is semantically secure against chosen plaintext attack.

5. Implementation

The proposed scheme is implemented on a laptop with 2.3GHz CPU and 5GB RAM. Implementation is done using NTL [11] with GMP using Cygwin tools on Windows 7 operating system. Dual RSA is also implemented with the same configuration for better analysis. All the algorithms, key generation, encryption, decryption are run 100 times and the time (in ms) is recorded by calculating the average. As the most popular modulus size is 1024 bits and 2048, the scheme is run by using both the sizes and performance is recorded.

Table 1 shows the time taken by key generation, encryption and decryption algorithms by the proposed scheme for n=1024 and n=2048 bits. The value was recorded for different values of n_d. In each record encryption time comes out as negligible.

With moduli 1024 bits, the following values are recorded:

n_1=1024 bits

n_2=1024 bits

n_d=342 bits

n_e=1024 bits

N1=11143337828858814366486156618227338718869534
22621130804339944857653295369784605319480473876577595258859839327259072828536405445112300739846149058889928389345802523761348743914261982639839512230909015544412

406398173158358889991455501957019508037542481271792552
751113948560083341440813638510119181415283455367 7

N2=165520943686534301068964697249010423696727469251984401647879533042344337969351518789678294233056383001087829926282681533273263946433951361316529741207166311978905713576196141730414176071883116480593799331363838376565216950518625635081265823842611895431291133196171447985665971359850417584242492652867114392499

Table 1. Implementation Results of the proposed scheme

n_d	Key-gen Time(ms)	Encryption Time(ms)	Decryption Time(ms)
N=1024 bits			
342	312	0.0109	1.24
350	390	0.0031	1.25
400	327	0.0234	1.56
450	343	0.0047	1.71
500	358	0.0046	1.88
N=2048 bits			
684	4531	0.031	8.27
700	4166	0.016	8.43
800	3999	0.016	9.83
900	4163	0.015	10.77
1000	3157	0.016	12.17

e=157621775754591269738556719887336905672337342368514384347904612952967416147501230142819781455468672801711983962630877460629605739946598376556249405190677928431157337800469140332740481156365599619537286936035724818873925444056830424062583445091377986115193303995058163024278890439269701112033638547829950033327

d=894214583801872834743653745327188985129635735782298100041069585299004582874552344355042251104367582 1743

The algorithm is also implemented with moduli 2048 bits and the following values were recorded:

n_1=2047 bits

n_2=2048 bits

n_d=684 bits

n_e=2047 bits

Table II shows the comparison of the proposed scheme and other RSA variants.

Table II. Comparison of the variants

RSA Variants	Encryption Time (ms)	Decryption Time(ms)
Basic RSA	0.13	3.9
RSA CRT	0.129	1.17
Dual RSA Small-d	3.9	1.4
Proposed Scheme	0.012	1.4

As shown in Table II, the encryption time comes out to be negligible as compared to Dual RSA Small-d. Here the comparison results are shown for the modulus size 1024 bits.

5. Conclusions

In this work the most popular public key cryptosystem, RSA, is improved. The improvement is done in Dual RSA Small d, so that the high computations involved towards encryption side are lower down. The proposed scheme is shown to get the improved variant of RSA. The implementation results show that besides less memory consumption, the proposed scheme is efficient in both encryption and decryption sides. The proposed scheme can be efficiently used for saving memory as well as computation cost. The scheme has its application in any area where the workload on the system is not balanced or in any memory constrained device.

References

[1] R. Rivest, A. Shamir and L. Adleman., "A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, 1978; 21(2):120-126.

[2] J J Quisquater and C Couvreur., "Fast decipherment algorithm for RSA public-key cryptosystem", Electronic Letters, 1982; 18:905–907.

[3] A. Fiat, "Batch RSA", Advances in Cryptology, 1989; 435:175–185.

[4] T. Collins, D. Hopkins, S. Langford and M. Sabin, "Public key cryptographic apparatus and method", US Patent #5, 1997; 848;159.

[5] T. Takagi, "Fast RSA-type cryptosystem modulo $p^k q$", Crypto'98, 1998;1462:318–326.

[6] M. Wiener, "Cryptanalysis of short RSA secret exponents", IEEE Transactions on Information Theory, 1990; 36(3):553–558.

[7] H.M. Sun, M.J Hinek and Wu ME, "Trading decryption for speeding encryption in Rebalanced-RSA", Journal of Systems and Software, 2009; 82(9):1503-1512.

[8] S.D.Galbraith, C. Heneghan, J.F McKee, "Tunable balancing of RSA", In Proceedings of ACISP'05, 2005; 3574:280–292.

[9] A.K.Lenstra, BM De Weger, "Twin RSA", Progress in Cryptology-MyCrypt 2005; 3715:222–228.

[10] H.M Sun, M.E Wu, W.CTing and M.J Hinek, "Dual RSA and its security analysis", IEEE Transactions on Information Theory, 2007; 53(8):2922-2933.

[11] V. Shoup, NTL: A Library for doing number theory. 2008, version 5.3.1. <http://shoup.net/ntl/>

[12] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption- How to Encrypt with RSA", In Eurocrypt '94, LNCS 950, 1995, 92-111.

[13] S. Golwasser, M. Micali, "Probabilistic Encryption", Journal of Computer and System Sciences, 1984; 28:270-299.

[14] D. Pointcheval, "New public key cryptosystem based on the dependent RSA problem", Eurocrypt'99 Springer-Verlag, 1999; 1592:239-254.

[15] S. Padhye, "On DRSA public key cryptosystem", International Arab Journal of Information Technology, 2006; 3(4):334-336.

[16] M. Bellare, P. Rogaway, "Random Oracles are Practical: a Paradigm for designing efficient protocols", in Proceedings of the 1st CCCS, ACM Press, 1993; 62-73.

[17] M.J Hinek, "On the security of some variants of RSA", Ph.D. thesis, University of Waterloo, 2007.

[18] S. Sarkar, S. Maitra, "Cryptanalysis of Dual CRT-RSA", IACR Cryptology eprint 2010

[19] S. Sarkar and S. Maitra, "Cryptanalytic results on Dual CRT-RSA and Common Prime RSA", Journal of Design and Codes Cryptography, 2013; 66:157-174.